PRE-APPEAL BRIEF REQUEST FOR REVIEW	Docket No. C0012/7000

Applicant:	Raju C. Bopardikar, Jack J. Stiffler, Jacob Y. Bast, Gary A. Cardone, David E. Kaufman, Stuart P. MacEachern, Bruce D. McLeod, James M. Nolan, Jr., Zdenek Radouch and James A. Wentworth, III
Serial No:	09/608,521
Filed:	June 30, 2000
For:	Method And Apparatus For Implementing High-Performance, Scaleable Data Processing And Storage Systems
Examiner:	A. M. Mirza
Art Unit:	2145

Mail Stop AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Applicant requests review of the final rejection in the above-identified application. No amendments are being filed with this request.

This request is being filed with a notice of appeal

The review is requested for the reason(s) stated on that attached sheet(s).

Respectfully submitted

/paul e. kudirka/ Date: 2008-03-26

Paul E. Kudirka, Esq. Reg. No. 26,931
LAW OFFICES OF PAUL E. KUDIRKA
Customer Number 64967
Tel: (617) 357-0010 Fax: (617) 357-0035

REASONS FOR REVIEW

Claims 1-11 and 44-54 have been rejected under 35 U.S.C. §103(a) as obvious over U.S. Patent No. 6,351,775 (Yu) in view of U.S. Patent No. 6,157,927 (Schaefer.) In conventional data storage systems that were available at the time that the application was filed, an address that identified the data directly corresponded to an actual physical location at which the data was to be stored. Therefore, during a typical read-modify-write sequence, data identified by an address was retrieved from a location specified by that address, modified and then stored back to the same location as specified by that address.

The present invention broke the correspondence between data identification and the location at which the data is stored. In particular, in the present system, a client wishing to retrieve or store data specifies a data identifier instead of a traditional address that is associated with a location. This data identifier is presented to an access interface which determines a location at which the data will be stored, based not on the identity of the data, but on other criteria such as the workload of each resource in the resource pool. The result of this operation is that data may be stored in a storage resource other than the storage resource from which that data was retrieved. Therefore, during a typical read-modify-write operation, data corresponding a data identifier would be read from a storage location and modified, but might be stored back into a location different from which it was read depending on resource load and availability. The data thus dynamically moves based on criteria determined by the data storage system. Consequently, load balancing becomes automatic.

The Yu reference is similar to the Lumelsky reference, previously cited. Both of these references disclose client-server systems in which clients request data from a pool of servers and receive data from a selected one of those servers. In these systems, duplicate data is stored on a plurality of servers in the pool and the requested data is retrieved from one of those servers. The server from which the data is retrieved is selected to achieve a goal such as load balancing. The goals are achieved by using criteria to route the data request (the Yu reference calls the data request a "data request object") generated by the client to one of the servers, which then services the request. However, once the server has been selected, the direction of data flow is always from

server to client. No data is received from clients and transferred to the servers and stored thereon – no storage write operations are performed. This is evident from the Yu abstract which discloses "In the Internet, the collection of servers can be either a proxy or Web server cluster and can include a DNS and/or TCP-router." and at Yu, column 5, line 67 to column 6, line 9, which state "The server cluster includes multiple server nodes (161-163) to handle high traffic demand. It can be either a proxy server or a Web server cluster. The servers in the cluster can include, but are not limited to, products such as are sold by IBM under the trademarks S/390 SYSPLEX, SP2, or RS6000 workstations. As is conventional, each request can be handled by any server in the cluster. Typical service requests include World-Wide-Web page accesses, remote file transfers, electronic mail, and transaction support."

The examiner points to Yu, column 4, lines 38-43 as disclosing data requests which include a data identifier and data to be stored received from the client. However, this section of Yu discloses that metadata is "piggybacked" on each data request and used to determine to which server the request is routed. The metadata is not received from the client and is not stored on the server. The examiner further refers to Yu, column 9, lines 51-67, as disclosing dynamically selecting a subset of the plurality of resources to which the data is transferred to be stored. However, this section of Yu only discloses how statistical criteria are used to determine to which server the client request is sent. As discussed above, once the request is routed to a particular server, the data flow is from the server to the client.

The Schaefer reference has been discussed in detail in previous responses. It discloses a conventional database transaction processing in which information is retrieved from, and stored back into, the same database location. Specifically, the storage location is determined by the address and not by the workload of the storage resources.

Therefore, neither of the cited references discloses a system in which data corresponding to the same data identifier may be stored in different storage locations based on criteria, such as load balancing or availability and, consequently, the combination of these references cannot teach or suggest this type of operation. The claims have been amended to particularly point out the differences between the present

system and the cited references. For example, claim 1 now recites, in lines 3-10, "…an access interface module which receives data storage requests, each including a data identifier, and data to be stored from the client and, in response to each service request and based on a workload instead of a location in the plurality of resources, dynamically selects a subset of the plurality of storage resources to which the data is transferred to be stored so that the location to which data corresponding to the same data identifier is transferred can change from request to request in order to dynamically distribute the workload across the plurality of storage resources…" (emphasis added). As discussed above, in the Yu system, no data storage operations are disclosed. In the Schaefer system, the resources to which data is transferred are selected by address not by workload as claimed. Thus, the combination of Yu and Schaefer references proposed by the examiner cannot teach or suggest the structure recited in claim 1 because neither references teaches this structure. Therefore, claim 1 patentably distinguishes over the cited reference combination.

Claims 2-11 are dependent, either directly or indirectly, on amended claim 1 and incorporate the limitations thereof. Therefore, they also distinguish over the cited reference combination in the same manner as amended claim 1.

Method claim 44 contains limitations that parallel those in apparatus claim 1 and has been amended in a manner that parallels that of claim 1. Thus, amended claim 44 patentably distinguishes over the cited reference combination in the same manner as claim 1. Claims 45-54 are dependent, either directly or indirectly, on amended claim 44 and incorporate the limitations thereof. Therefore, they also distinguish over the cited reference combination in the same manner as amended claim 44.